MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 985 October, 1987

On the Recognition of Parameterized Objects

W. Eric L. Grimson

MIT Artificial Intelligence Laboratory
545 Technology Square
Cambridge, Mass. 02139

Abstract. Determining the identity and pose of occluded objects from noisy data is a critical step in interacting intelligently with an unstructured environment. Previous work has shown that local measurements of position and surface orientation may be used in a constrained search process to solve this problem, for the case of rigid objects, either two-dimensional or three-dimensional. This paper considers the more general problem of recognizing and locating objects that can vary in parameterized ways. We consider objects with rotational, translational or scaling degrees of freedom, and objects that undergo stretching transformations. We show that the constrained search method can be extended to handle the recognition and localization of such generalized classes of object families.

Acknowledgements: This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the System Development Foundation, in part by an Office of Naval Research University Research Initiative grant under contract N00014-86-K-0685, in part by the Advanced Research Projects Agency of the Department of Defense under Army contract number DACA76-85-C-0010 and in part by DARPA under Office of Naval Research contract N00014-85-K-0124.

1. Introduction

The general problem considered in this note is how to locate a known object from sensory data, especially when that object may be occluded by other (possibly unknown) objects. In previous work [Grimson and Lozano-Pérez 84, 87] we described a recognition system, called **RAF** (for **R**ecognition and **A**ttitude **F**inder), that identifies and locates objects from noisy, occluded data. In that work, we concentrated on a particular subclass of rigid models. If the sensory data provided two-dimensional geometric data, for example intensity edges from a visual image, we considered the recognition of objects that consisted of sets of linear segments, or equivalently, polygonal objects in which some edges are not included. If the sensory data was three-dimensional, we considered the recognition of objects that consisted of sets of planar fragments, or equivalently, polyhedral objects in which some of the faces are not included.

In general, of course, we cannot guarantee that the recognition system will be confronted only with rigid polyhedral objects of known size. The **RAF** has been extended to deal with curved objects, in the two-dimensional case [Grimson 1987]. In this note, we consider extensions of our method to deal with families of objects that are characterized by sets of free parameters.

2. Recognition as constrained search

Before dealing with the problem of parameterized parts, we briefly review the recognition method used [Grimson and Lozano-Pérez 84, 87].

2.1 Definition of a solution

Suppose we are given a set of data fragments, obtained from the boundary of an object or objects, and measured in a coordinate system centered about the sensor. Suppose we are also given a set of object models, specified by a set of faces (whose definition we will make formal shortly) measured in a local coordinate frame specific to the model. A solution to the recognition problem consists of a three-tuple

$$\langle \texttt{object}_i, \{(d_{i_1}, m_{j_1}), (d_{i_2}, m_{j_2}), \ldots (d_{i_k}, m_{j_k})\}, (R, \mathbf{v_0}) \rangle$$

where $\texttt{object}_i$ identifies which object from a library of known objects, the d, m pairings are associations of a subset of the sensory data d with model faces m from $\texttt{object}_i$ and R is a rotation matrix, and $\mathbf{v_0}$ is a translation vector such that a vector $\mathbf{v}_m$ in model coordinates is transformed into a vector $\mathbf{v}_d$ in sensor coordinates by

$$\mathbf{v}_d = R\mathbf{v}_m + \mathbf{v_0}$$

and where this coordinate frame transformation maps the model from its local coordinate frame into the sensor coordinate frame in such a manner that each data fragment correctly lies on its assigned model face.

As has been described elsewhere [Grimson and Lozano-Pérez 84, 87], we approach the recognition problem as one of search. Thus, we first focus on finding legitimate pairings of data and model fragments, for some subset of the sensory data. We chose to structure this search process as a constrained depth first search, using an *interpretation tree* (IT). Each node of the tree describes a partial interpretation of the data, and implicitly contains a set of pairings of data fragments and model faces. Nodes at the first level of the tree define assignments for the first data fragment, nodes at the second level define assignments for the first and second data fragments, and so on. Each node branches at the next level in up to $n + 1$ ways, where n is the number of model faces in the object. The last branch is a *wild card* or *null* branch and has the effect of excluding the data fragment corresponding to the current level of the tree from part of the interpretation.

Given s data fragments, any leaf of the tree specifies an interpretation

$$\{(d_1, m_{j_1}), (d_2, m_{j_2}), \ldots (d_s, m_{j_s})\},$$

where some of the m_{j_k} may be the wild card character. By excluding such matches, the leaf yields a partial interpretation

$$\left\{(d_{i_1}, m_{j_{i_1}}), (d_{i_2}, m_{j_{i_2}}), \ldots (d_{i_k}, m_{j_{i_k}})\right\}$$

where $1 \leq i_1 < i_2 < \ldots < i_k$ but these indices may not include the entire set from 1 to s. This interpretation may then be used to solve for a rigid, scaled transformation that maps model faces into corresponding data fragments, if such a transformation exists. Thus, by searching for leaves of the tree and testing that the interpretation there yields a legal transformation, we can find possible instances of object models in the data.

Since this search process is inherently an exponential problem, the key to an efficient solution is to use constraints to remove large subtrees from consideration without having explicitly to explore them. In [Grimson and Lozano-Pérez 84, 87] we describe a set of constraints based on the local shape of parts of objects, either in two dimensions or in three. In this work, the object models and the sensory data consist of linear edge or face fragments. The constraints include the following:

- The length of a data fragment must be smaller than the length of a corresponding model fragment, up to some bounded measurement error;
- The angle between the normals to a pair of data fragments must differ from the angle between the normals of the corresponding model fragments by no more than a bounded measurement error;
- The range of distances between two data fragments must lie within the range of distances of the corresponding model fragments, where the model range has been expanded to account for measurement errors;
- The range of components of a vector spanning the two data fragments in the direction of each of the fragment's normal must lie within the corresponding range

of components for vectors spanning the model fragments, modulo measurement error.

In [Grimson 87] we extended these constraints to include curved objects in two dimensions.

2.2 The constraints reduce the search

Given these unary and binary constraints, the constrained search process can be straightforwardly specified. Suppose the search process is currently at some node at level k in the interpretation tree and with a consistent partial interpretation given by

$$\{(d_1, m_{j_1}), (d_2, m_{j_2}), \ldots (d_k, m_{j_k})\}.$$

We now consider the next data fragment d_{k+1}, and its possible assignment to model face $m_{j_{k+1}}$, where j_{k+1} varies from 1 to $n+1$.

The following rules hold.

- If $m_{j_{k+1}}$ is the wild card match, then the new interpretation
$$\{(d_1, m_{j_1}), (d_2, m_{j_2}), \ldots (d_{k+1}, m_{j_{k+1}})\}$$
is consistent, and we continue downward in our search.
- If $m_{j_{k+1}}$ is a real model edge segment, we must verify that the length constraint holds for matching d_{k+1} to $m_{j_{k+1}}$, and that the angle, distance and component constraints hold for the pairings $(d_{k+1}, m_{j_{k+1}})(d_i, m_{j_i})$, for $1 \leq i \leq k$.
- If all of these constraints are true, then
$$\{(d_1, m_{j_1}), (d_2, m_{j_2}), \ldots (d_{k+1}, m_{j_{k+1}})\}$$
is a consistent partial interpretation, and we continue our depth first search. If one of them is false, then the partial interpretation is inconsistent. In this case, we increment the model face index j_{k+1} by 1 and try again, until $j_{k+1} = n+1$.

If the search process is currently at some node at level k in the interpretation tree, and has an inconsistent partial interpretation given by

$$\{(d_1, m_{j_1}), (d_2, m_{j_2}), \ldots (d_k, m_{j_k})\}$$

then it is in the process of backtracking. If $j_k = n+1$ (the wild card) we backtrack up another level, otherwise we increment j_k and continue.

2.3 Model tests

Once the search process reaches a leaf of the interpretation tree, we have accounted for all of the data points. We are now ready to determine if the interpretation is in fact globally valid. To do this, we solve for a rigid transformation mapping points $\mathbf{v}_m$ in model coordinates into points $\mathbf{v}_d$ in sensor coordinates,

$$\mathbf{v}_d = R\mathbf{v}_m + \mathbf{v}_0$$

where R is a rotation matrix, and $\mathbf{v}_0$ is a translation vector. We can solve for this transformation in a number of ways [e.g. Grimson and Lozano-Pérez 84, 87, Ayache and Faugeras 86].

Given such a transformation, which is usually found using some type of least squares fit, we must then ensure that the interpretation actually satisfies it. We do this by considering each of the data fragments associated with a real model fragment in the interpretation, and transforming the associated model fragment by the computed transform. For each such fragment, we then verify that the transformed fragment differs in position and orientation from its associated data fragment by amounts that are less than some acceptable error bounds. These bounds on transform error can be obtained from the predefined bounds on the sensor error [Grimson 86b]. Any interpretation that passes such a model test is a consistent interpretation of the data.

2.4 Additional search reductions

While the constrained search technique described above will succeed in finding all consistent interpretations of the sensory data, for a given object model, it is not particularly computationally efficient. This is mostly due to the problem of segmenting the data to determine subsets that belong to a single object. Indeed, if all of the sensory data do belong to one object, the described method is known to be quite efficient, as has been verified both empirically [Grimson and Lozano-Pérez 84, 87] and theoretically [Grimson 1986a]. In order to improve the efficiency of the method, we can add two additional methods to our search process, both previously discussed for the case of linear fragments in [Grimson and Lozano-Pérez 87], and extended to circular segments in [Grimson 87].

The first is to use a parameter hashing scheme, such as a Hough transform, to hypothesize small subspaces of the entire search space that are likely to contain an interpretation. The second is to use a measure of matching, such as the portion of the object perimeter correctly accounted for by the matched sensory data, to prematurely terminate the search process. In the work described here, we use only the second heuristic.

3. Parameterized Families

3.1 Examples of Parameterized Objects

While our previous work has illustrated the utility of our approach to the problem of rigid objects, we are interested here in extending the method to deal with parameterized objects. We consider a number of different possibilities.

Scale

Perhaps the simplest example of a parameterized family is that defined by a rigid object that can undertake a range of possible sizes, that is, the shape of the object is fixed, but the overall scale factor can vary. Many techniques for object recognition and localization can easily deal with this case, since the scale factor can simply be considered part of the coordinate frame transformation required to map the model patches into their corresponding sensed patches.

Coordinate-frame transformations

A more interesting class of parameterized objects are those that involve a limited number of moving parts. A good example is a pair of scissors, which has a single degree of freedom, namely the rotation of the two blades relative to a common joint. We would like to be able to recognize the scissors, independent of the relative orientation of the blades, and without requiring a different model to represent each orientation. This class could further be extended to include scissors of different sizes.

Stretching deformations

A third class of parameterized objects are those in which subparts can stretch along an axis. An example would be a family of hammers, for which there is a generic handle shape, but which can stretch along the axis of the handle, as indicated in Figure 1.

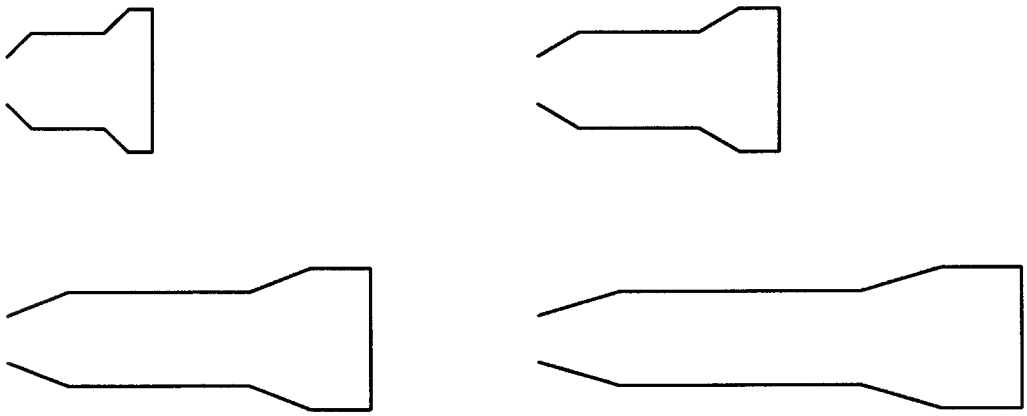

Figure 1. A set of parameterized subparts, in which the generic shape in the upper left is stretched along the axis of the shape.

Our goal is to extend our recognition method to handle such classes of parameterized families.

In dealing with parameterized families, we restrict our attention to two dimensional objects, that are composed of sets of linear edge fragments. Each **linear edge fragment** consists of two endpoints, and a unit vector normal to the line between them and pointing away from the interior of the object. Formally, this is given by

$$\texttt{linear}_i = (\hat{\mathbf{n}}_i, (\mathbf{b}_i, \mathbf{e}_i)) .$$

Note that a point on the edge can be represented by

$$\hat{\mathbf{n}}_i \qquad \text{and} \qquad \mathbf{b}_i + \alpha_i \hat{\mathbf{t}}_i, \quad \alpha_i \in [0, \ell_i]$$

where $\hat{\mathbf{n}}_i$ is the unit normal vector, $\hat{\mathbf{t}}_i$ is a unit tangent vector, oriented so that it points from $\mathbf{b}$ to $\mathbf{e}$, and α_i can vary from 0 to the length of the edge ℓ_i.

3.2 Possible approaches

A large number of methods have been explored in the literature for recognizing rigid objects, both in two dimensions and in three. When considering parameterized objects, far fewer methods have been considered. In particular, while a number of schemes have been suggested for representing parameterized objects, such as generalized cylinders and superquadrics, at this point very few actual recognition engines based on such parameterized representations have been demonstrated. The best known such system is probably ACRONYM [Brooks 1981]. Within the context of our approach to recognition, there are two distinct alternatives for extending the method to handle parameterized parts, both related in a global sense to the approach taken by Brooks.

The first approach is to extend our geometric constraints to directly incorporate the free parameters. In this case, the search process would become a constraint propagation technique, in which the current range of possible values for each of the parameters would be passed from a parent node of the IT to each of its sons. At each new node, the constraints imposed by matching the new data patch to its assigned model patch would be used to refine the range of free parameters, which would then be passed to that node's children. If any parameter is reduced to an empty range of values, the interpretation is inconsistent and the search along that subtree can be terminated.

The main difficulty with this approach is finding a clean way of representing the parameterized constraints, especially in a manner that will easily allow the computing and updating of feasible ranges for each of the parameters. Consider our example of a pair of scissors, where the parameter to be determined is the angle between the two blades. If two data fragments are being considered as belonging to two model fragments that are part of the same rigid subpart, then the constraints are the same as in our earlier approach. They either indicate consistency, in which case the range of possible values for the rotation parameter remains the same as it was before considering this pairing, or they indicate inconsistency, in which case the search must backtrack. On the other hand, suppose two data fragments are being considered as belonging to model fragments on different rigid subparts. In this case,

we need a means of expressing the range of possible values for the rotation parameter as an explicit function of the relative geometry of the two model fragments and the two data fragments. This may prove difficult to obtain.

The second approach is to break the object model into rigid subparts, all of which are connected to a global model-based coordinate frame through a series of coordinate frame transformations. Each subpart can then be recognized by application of our earlier technique, including a free scale parameter. Once the subparts have been recognized and located, we must check that they are consistent by confirming that the parts satisfy a set of predetermined global coordinate-frame constraints.

Consider the earlier example of a pair of scissors, with a free overall scale factor. In this approach, we treat each blade of the scissors as a rigid subpart. Thus, we attempt to locate instances of the right and left blade in the sensory data. Once we have done this, we then confirm that the subparts are parts of a consistent whole. In the case of the scissors, this would involve checking two things: (1) the scale factor associated with each blade is roughly the same, and (2) the transformations from model coordinates to sensory coordinates associated with each blade are such that the position, in sensor coordinates, of the pin joining the two blades is roughly the same (i.e. the located instances of the blades in the data are rotated about the expected common axis). The advantage of this second method is that the geometric constraints remain simple, yet combinatorially powerful.

Note that we can apply our search for rigid subparts in several ways. The simplest is to search the data independently for each rigid subpart, then test all possible combinations of subparts for consistent wholes. A more efficient method would be to first search the data for one subpart (e.g. the largest). For each candidate solution found in the data, we can then use limits on the ranges of the parameters to restrict the possible positions of the other subparts in the sensory data. Using this reduced data set, we can then search for instances of the other subparts, testing each instance for global consisistency. If no instance of the initial seed subpart is found, (for example, it is occluded in the data) we can then consider the next seed subpart (e.g. the next largest) and proceed as before.

In this paper, we explore both options. We first derive the set of geometric constraints on interpretations, and then illustrate the search process on some simple examples.

3.3 Scale Factors

Perhaps the simplest family of objects to consider are those in which a single, rigid object of known shape can undergo an arbitrary global scaling, within some limits. We need to consider how to adjust the recognition process, so that it can not only recognize where an object is in the data, but also its overall size.

We assume that the scale factor is applied to the data, so that the transformation from a point in model coordinates, $\mathbf{v}_m$, to sensor coordinates, $\mathbf{v}_d$, is given

by

$$s\mathbf{v}_d = R_\theta \mathbf{v}_m + \mathbf{v}_0$$

where s is a scale factor, θ is an angle, R_θ is a rotation matrix of angle θ and $\mathbf{v}_0$ is a translation vector.

3.3.1 Length constraint

If we are matching data edge d_i, given by

$$(\hat{\mathbf{n}}_i, (\mathbf{b}_i, \mathbf{e}_i))$$

to model edge m_p, given by

$$\left(\hat{\mathbf{N}}_p, (\mathbf{B}_p, \mathbf{E}_p)\right)$$

then the length of the data edge must be less than the length of the corresponding model edge, modulo measurement error. We let ℓ_i denote the length of the data fragment, and L_p denote the corresponding length of the model fragment, where these lengths are given by

$$\ell_i = |\mathbf{b}_i - \mathbf{e}_i|, \qquad L_p = |\mathbf{B}_p - \mathbf{E}_p|.$$

Then we must have

$$s\ell_i \leq L_p + \epsilon_l \qquad \forall s$$

where ϵ_L is a predefined upper bound on the amount of error inherent in measuring the length of an edge.

We can define

$$\textbf{scaled-length-constraint}(i,p) = \left[0, \frac{L_p + \epsilon_L}{\ell_i}\right]$$

that is, the range of scales consistent with this assignment. This constraint returns a (possibly empty) range of values.

3.3.2 Angle constraint

Let θ_{ij} denote the angle between $\hat{\mathbf{n}}_i$ and $\hat{\mathbf{n}}_j$, and let Θ_{pq} denote the angle between $\hat{\mathbf{N}}_p$ and $\hat{\mathbf{N}}_q$. We let

$$\textbf{binary-angle-constraint}(i,j,p,q) = \text{True iff } \theta_{ij} \in [\Theta_{pq} - 2\epsilon_a, \Theta_{pq} + 2\epsilon_a]$$

where all arithmetic comparisons are performed modulo 2π and where ϵ_a is an upper bound on the amount of error inherent in determing the direction of a normal.

3.3.3 Component constraint

The third constraint concerns the separation of the two edge fragments. In particular, we consider the range of components of a vector between the two edge fragments, in the direction of each of the edge normals. Algebraically, this is expressed by the dot product

$$\langle \mathbf{b}_i + \alpha_i \hat{\mathbf{t}}_i - \mathbf{b}_j - \alpha_j \hat{\mathbf{t}}_j, \hat{\mathbf{n}}_i \rangle$$

which reduces to

$$\langle \mathbf{b}_i - \mathbf{b}_j, \hat{\mathbf{n}}_i \rangle - \alpha_j \langle \hat{\mathbf{t}}_j, \hat{\mathbf{n}}_i \rangle \qquad \alpha_j \in [0, \ell_j]$$

Of course, there is an equivalent constraint for components in the direction of $\hat{\mathbf{n}}_j$. Note that this expression actually determines a range of values, with extrema when $\alpha_j = 0, \ell_j$. We denote this by

$$d^{\perp}_{\ell,ij} = \min\{\langle \mathbf{b}_i - \mathbf{b}_j, \hat{\mathbf{n}}_i \rangle - \alpha_j \langle \hat{\mathbf{t}}_j, \hat{\mathbf{n}}_i \rangle \,|\alpha_j \in \{0, \ell_j\}\}$$
$$d^{\perp}_{h,ij} = \min\{\langle \mathbf{b}_i - \mathbf{b}_j, \hat{\mathbf{n}}_i \rangle - \alpha_j \langle \hat{\mathbf{t}}_j, \hat{\mathbf{n}}_i \rangle \,|\alpha_j \in \{0, \ell_j\}\}$$

These ranges can be computed both for pairs of data edges and pairs of model edges. In the ideal case, consistency will hold only if the data range is contained within the model range (since the data edges may correspond to only parts of the model edges). As in the case of the other constraints, we also need to account for error in the measurements. We derive a simple method for doing this below.

Consider the base case, shown in Figure 2a. The perpendicular distance from the endpoint of one edge to the other edge is shown as $D^{\perp}$. In Figure 2b, the edge is rotated by ϵ_a about its midpoint, and the new perpendicular distance X is shown. We need to relate X to measurable values. We already have $D^{\perp}$. We can also measure S, the distance from the midpoint of the edge to the perpendicular dropped from the endpoint of the other edge, as shown. Straightforward trigonometry then yields the new distance

$$X = \left(D^{\perp} - S \sin \epsilon_a\right) \cos \epsilon_a.$$

Since the position of the second edge is not known exactly, we must adjust this expression, to yield one limit on the range of possible measurements:

$$D^{\perp}_{\ell,pq} = \left(D^{\perp} - S \sin \epsilon_a\right) \cos \epsilon_a - \epsilon_p.$$

The other extreme is shown in Figure 2c. Trigonometric manipulation yields the following upper bound

$$D^{\perp}_{h,pq} = \left(S - D^{\perp} \sin \epsilon_a\right) \sin \epsilon_a + D^{\perp} \sec \epsilon_a + \epsilon_p.$$

Thus, given two model edges indexed by p, q, we can compute a range of possible measurements (modulo known error bounds), by using $D^{\perp}_{\ell,pq}$ and $D^{\perp}_{h,pq}$ computed over all the endpoints of the edges. We denote this range by $[M^{\perp}_{\ell,pq}, M^{\perp}_{h,pq}]$.

Given a range of projections of data edge i onto edge j, and a corresponding range of projections for model edge p onto model edge q, we need to determine bounds on s such that

$$[sd^{\perp}_{\ell,ij}, sd^{\perp}_{h,ij}] \subseteq [M^{\perp}_{\ell,pq}, M^{\perp}_{h,pq}].$$

The following cases hold:

If $\langle \hat{\mathbf{t}}_i, \hat{\mathbf{n}}_j \rangle > 0$ then

If $\langle \mathbf{b}_j - \mathbf{b}_i, \hat{\mathbf{n}}_j \rangle > 0$

then $s_h \leq \dfrac{M^{\perp}_{h,pq}}{\langle \mathbf{b}_j - \mathbf{b}_i, \hat{\mathbf{n}}_j \rangle}$

If $\langle \mathbf{b}_j - \mathbf{b}_i, \hat{\mathbf{n}}_j \rangle < 0$

then $s_\ell \geq \dfrac{M^{\perp}_{h,pq}}{\langle \mathbf{b}_j - \mathbf{b}_i, \hat{\mathbf{n}}_j \rangle}$

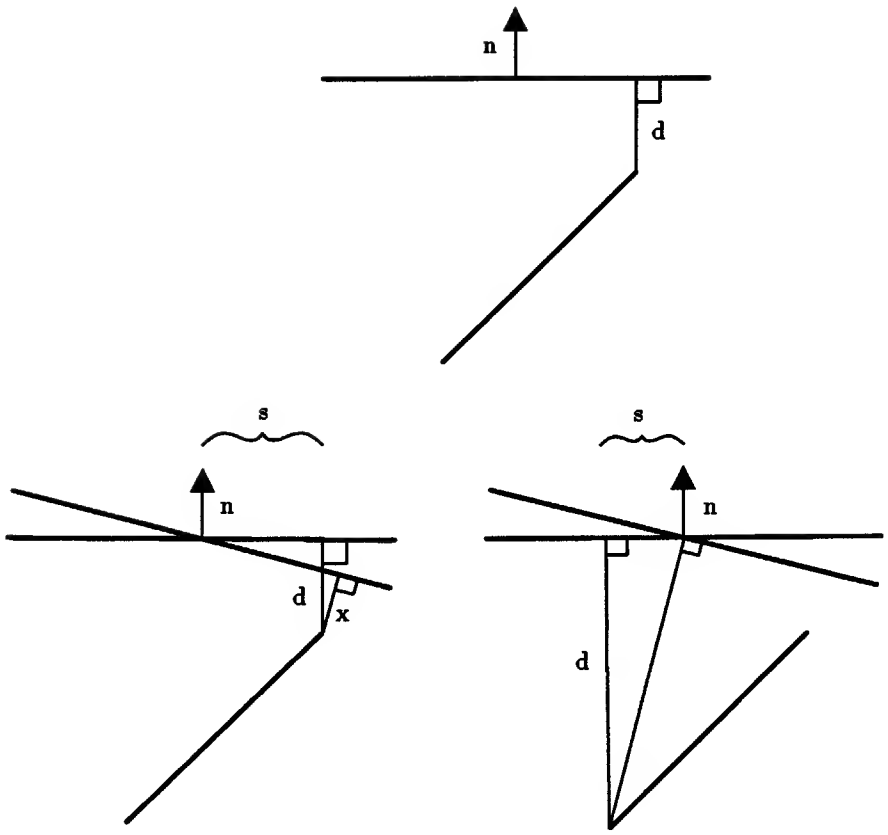

Figure 2. Errors in computing the direction constraint. (a) The component of a vector from one endpoint in the direction of the other edge's normal is given by the perpendicular distance **d** to the extended edge. (b) Since the actual normal is only accurate to within ϵ_a, one extreme case is given by rotating the extended edge about its midpoint by that amount and finding the new perpendicular distance. (c) The other extreme is obtained by considering the other endpoint.

$$\text{If} \quad \langle \mathbf{b}_j - \mathbf{b}_i, \hat{\mathbf{n}}_j \rangle - \ell_i \langle \hat{\mathbf{t}}_i, \hat{\mathbf{n}}_j \rangle > 0$$

$$\text{then} \quad s_\ell \geq \frac{M^{\perp}_{\ell,\,pq}}{\langle \mathbf{b}_j - \mathbf{b}_i, \hat{\mathbf{n}}_j \rangle - \ell_i \langle \hat{\mathbf{t}}_i, \hat{\mathbf{n}}_j \rangle}$$

$$\text{If} \quad \langle \mathbf{b}_j - \mathbf{b}_i, \hat{\mathbf{n}}_j \rangle - \ell_i \langle \hat{\mathbf{t}}_i, \hat{\mathbf{n}}_j \rangle < 0$$

$$\text{then} \quad s_h \leq \frac{M^{\perp}_{\ell,\,pq}}{\langle \mathbf{b}_j - \mathbf{b}_i, \hat{\mathbf{n}}_j \rangle - \ell_i \langle \hat{\mathbf{t}}_i, \hat{\mathbf{n}}_j \rangle}$$

If $\langle \hat{\mathbf{t}}_i, \hat{\mathbf{n}}_j \rangle < 0$ then

$$\text{If} \quad \langle \mathbf{b}_j - \mathbf{b}_i, \hat{\mathbf{n}}_j \rangle > 0$$

$$\text{then} \quad s_\ell \geq \frac{M^{\perp}_{\ell,\,pq}}{\langle \mathbf{b}_j - \mathbf{b}_i, \hat{\mathbf{n}}_j \rangle}$$

$$\text{If} \quad \langle \mathbf{b}_j - \mathbf{b}_i, \hat{\mathbf{n}}_j \rangle < 0$$

$$\text{then} \qquad s_h \leq \frac{M^{\perp}_{\ell,\,pq}}{\langle \mathbf{b}_j - \mathbf{b}_i,\, \hat{\mathbf{n}}_j \rangle}$$

$$\text{If} \quad \langle \mathbf{b}_j - \mathbf{b}_i,\, \hat{\mathbf{n}}_j \rangle - \ell_i \langle \hat{\mathbf{t}}_i,\, \hat{\mathbf{n}}_j \rangle > 0$$

$$\text{then} \qquad s_h \leq \frac{M^{\perp}_{h,\,pq}}{\langle \mathbf{b}_j - \mathbf{b}_i,\, \hat{\mathbf{n}}_j \rangle - \ell_i \left\langle \hat{\mathbf{t}}_i,\, \hat{\mathbf{n}}_j \right\rangle}$$

$$\text{If} \quad \langle \mathbf{b}_j - \mathbf{b}_i,\, \hat{\mathbf{n}}_j \rangle - \ell_i \langle \hat{\mathbf{t}}_i,\, \hat{\mathbf{n}}_j \rangle < 0$$

$$\text{then} \qquad s_\ell \geq \frac{M^{\perp}_{h,\,pq}}{\langle \mathbf{b}_j - \mathbf{b}_i,\, \hat{\mathbf{n}}_j \rangle - \ell_i \left\langle \hat{\mathbf{t}}_i,\, \hat{\mathbf{n}}_j \right\rangle}$$

Thus, based on the measured and model constraint ranges, we can compute a range of scale factors $[s_{\ell,i,j,p,q}, s_{h,i,j,p,q}]$ for which the assignment of data edges to model edges is consistent. We let

$$\textbf{scaled-component-constraint}(i,j,p,q) = [s_{\ell,i,j,p,q}, s_{h,i,j,p,q}].$$

Although we will not use it here, note that we could derive a similar form for the distance constraint.

Given these unary and binary constraints, we can now modify our constrained search process. With each node of the search tree, we associate a range of consistent values for the scale parameter, which we will denote $[s_\ell^{(k)}, s_h^{(k)}]$, where k indicates the level of the node in the tree. Suppose the search process is currently at some node at level k in the interpretation tree and with a consistent partial interpretation given by

$$\{(d_1, m_{j_1}), (d_2, m_{j_2}), \ldots (d_k, m_{j_k})\}.$$

We now consider the next data fragment d_{k+1}, and its possible assignment to model fragment $m_{j_{k+1}}$, where j_{k+1} varies from 1 to $n+1$.

The following rules hold.

- If $m_{j_{k+1}}$ is the wild card match, then the new interpretation
$$\{(d_1, m_{j_1}), (d_2, m_{j_2}), \ldots (d_{k+1}, m_{j_{k+1}})\}$$
is consistent, and we continue downward in our search, setting
$$[s_\ell^{(k+1)}, s_h^{(k+1)}] = [s_\ell^{(k)}, s_h^{(k)}].$$
- If $m_{j_{k+1}}$ is a linear edge segment, we let
$$[s_\ell^{(k+1)}, s_h^{(k+1)}] = [s_\ell^{(k)}, s_h^{(k)}] \bigcap \textbf{scaled-length-constraint}(k+1, j_{k+1}).$$
If this new range is non-empty, then for all $i \in \{1, \ldots, k\}$ such that d_i is a linear edge fragment, we verify that
$$\textbf{binary-angle-constraint}(i, k+1, j_i, j_{k+1}) = \text{True}$$
and we set
$$[s_\ell^{(k+1)}, s_h^{(k+1)}] = [s_\ell^{(k+1)}, s_h^{(k+1)}] \bigcap \textbf{scaled-component-constraint}(i, k+1, j_i, j_{k+1}) \bigcap \textbf{scaled-component-constraint}(k+1, i, j_{k+1}, j_i).$$
- If
$$[s_\ell^{(k+1)}, s_h^{(k+1)}]$$

is non-empty, then

$$\{(d_1, m_{j_1}), (d_2, m_{j_2}), \ldots (d_{k+1}, m_{j_{k+1}})\}$$

is a consistent partial interpretation, and we continue our depth first search. Otherwise, the partial interpretation is inconsistent. In this case, we increment the model fragment index j_{k+1} by 1 and try again, until $j_{k+1} = n + 1$.

If the search process is currently at some node at level k in the interpretation tree, and has an inconsistent partial interpretation given by

$$\{(d_1, m_{j_1}), (d_2, m_{j_2}), \ldots (d_k, m_{j_k})\}$$

then it is in the process of backtracking. If $j_k = n + 1$ (the wild card) we backtrack up another level, otherwise we increment j_k and continue.

In this manner, we can naturally extend our constrained search method to recognize objects from families in which the free parameter is overall scale. An example is shown in Figure 3.

3.4 Rotating Subparts

More interesting classes of parameterized families include those in which parts of the object are allowed to move with respect to one another. A good example of such a family is a pair of scissors. A fixed size pair of scissors has a single degree of freedom, namely the rotation of the two blades relative to a common joint. We would like to be able to recognize the scissors, independent of the relative orientation of the blades, and without requiring a different model to represent each orientation. This class could further be extended to include scissors of different sizes.

As we suggested earlier, this could be done by generalizing the constraints to directly take the free parameters into account. However, an easier approach is to break the object up into rigid subparts, and deal with each separately. We illustrate this with our scissors example.

Suppose we treat each blade assembly as a separate part. We choose the location of the common pin as the origin of the model coordinate frame. Now suppose that we run our recognition system on each part, solving for a transformation $\theta_L, s_L, \mathbf{v}_{0,L}$ for the left blade and for a transformation $\theta_R, s_R, \mathbf{v}_{0,R}$ for the right blade. This can proceed in a manner identical to that described previously. To ensure that the two subparts are actually part of a common whole, we need to test that their interpretations are globally consistent. This can be done by means of a simple set of geometric constraints on their respective transformations. In this case, we require

$$s_L \approx s_R$$

$$\mathbf{v}_{0,L} \approx \mathbf{v}_{0,R}$$

Note that θ_L and θ_R could in principle take on any values. In practice, there is a limited range of orientations that the scissors can take on, so that a third constraint would be

$$\|\theta_L - \theta_R\| \leq C$$

Figure 3. Examples of recognition when the free parameter is overall scale. The first part shows a set of linear edges segments, the second shows the overlay of the located object, and the third shows the located object in isolation.

where C is some threshold on the range of rotations, and the arithmetic is done modulo 2π. An example is shown in Figure 4.

Note that the search can be done independently for each part, followed by the application of the global constraints on each candidate pair of subparts. More effectively, we can first solve for the location of one of the subparts, and then use

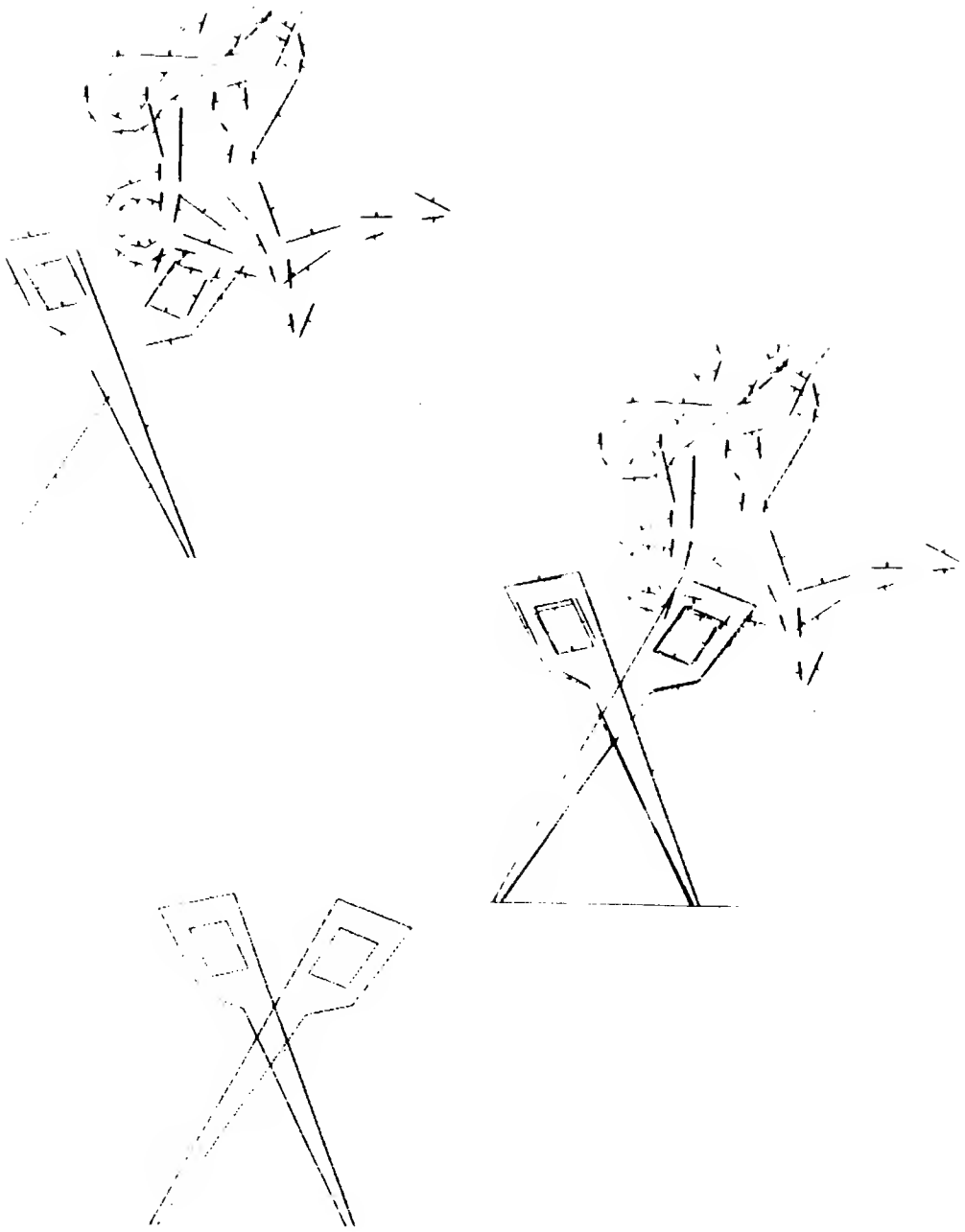

Figure 4. Examples of recognition when the free parameters are overall scale and rotation about a common axis. The first part shows a set of linear edges segments, the second shows the overlay of the located object, and the third shows the located object in isolation.

that position to restrict the possible positions of the second part, thereby directly removing some portions of the sensory data from consideration. We can also use the solution for the first subpart to restrict the values of the free parameters, for

example, limiting the range of acceptable scale factors before beginning the search for the second subpart.

We also need to add another level of backtracking search to our process. In particular, suppose we have found a candidate for the first rigid subpart, but we cannot find an acceptable candidate for the second one. In this case, we must backtrack to the point in the search for the first subpart at which we found the first candidate, and continue that search. If a second candidate for the first subpart can be found, then we can initialize a new search for the second subpart, and so on.

In this case, the data structures used to represent an object become somewhat more complex than in the case of rigid objects. Here, an object representation must include: a list of rigid subparts, each of which is represented by a set of constraint tables as in the original recognition method; a list of the free parameters; a set of procedures for verifying the post constraints; and a procedure for generating the restricted search area for a part, as a function of the pose of solution for other parts.

3.5 Subparts that Stretch

As a third example, consider a family of tools, say a set of hammers with identical heads, but different handles. Again, we would like to extend our method to recognize both the identity and location of the hammer, and to determine which handle is attached. To model the handles, we assume that a generic shape (such as that shown in the left of figure 1) can stretch by some variable amount along an axis (in the case of the handle in figure 1 this is the axis of symmetry). The problem is to extend the search method to deal with constraints that are themselves parameterized. We do this as follows.

Without loss of generality, we assume that the model part has been oriented so that the axis of stretching is the x-axis in model coordinates. We let α denote the amount of stretching along that axis, with $\alpha = 1$ designating the base case. Note that α is likely to be restricted to some range of values, which may be specified beforehand.

Consider first the constraints on the surface normals. In the case of rigid models, our constraint was that the angle between two data normals must be the same as the angle between the corresponding model normals, to within some error. In the case of stretching parts, the normals will vary relative to one another as a function of the stretching parameter α. Suppose we let θ_{ij} denote the measured angular difference, we let ϵ denote the allowed error range in measuring the angles, and we let Φ_p, Φ_q denote the corresponding model angles, in model coordinates, for the base case $\alpha = 1$. By appropriate algebraic manipulation, the following cases hold.

- $\Phi_p, \Phi_q \in \{0, \pi, \frac{\pi}{2}, -\frac{\pi}{2}\}$. In this case, we need only check that $\theta_{ij} \in [\Phi_p - \Phi_q - \epsilon, \Phi_p - \Phi_q + \epsilon]$.
- $\Phi_p \in \{0, \pi\}$. $\Phi_q \notin \{0, \pi, \frac{\pi}{2}, -\frac{\pi}{2}\}$. In this case, the stretching factor is given by

$$\alpha = \frac{-\tan\theta_{ij}}{\tan\Phi_q}.$$

A similar case holds when the roles of i and j are reversed.

- $\Phi_p \in \{\frac{\pi}{2}, -\frac{\pi}{2}\}$. $\Phi_q \notin \{0, \pi, \frac{\pi}{2}, -\frac{\pi}{2}\}$. In this case, the stretching factor is given by

$$\alpha = \frac{1}{\tan\theta_{ij} \tan\Phi_q}.$$

A similar case holds when the roles of i and j are reversed.

- $\tan\Phi_p \neq 0, \tan\Phi_q \neq 0, \tan\theta_{ij} = 0$ In this case, $\alpha = 0$ which indicates an inconsistency.
- All other cases. The stretching factor is given by

$$\alpha = \frac{\tan\Phi_p - \tan\Phi_q}{2\tan\Phi_p \tan\Phi_q \tan\theta_{ij}} \left[1 + \sqrt{1 - \frac{4\tan^2\theta_{ij} \tan\Phi_p \tan\Phi_q}{(\tan\Phi_p - \tan\Phi_q)^2}}\right].$$

Note that the measurement θ_{ij} is actually a range of measurements, due to error in sensory data. Thus, by applying the above computation over a sampling of values for θ_{ij} within this range, we can obtain a range of consistent values for the stretch factor, which we represent by

$$[\alpha_{\ell,i,j,p,q}, \alpha_{h,i,j,p,q}]$$

and we define

$$\textbf{stretch-angle-constraint}(i,j,p,q) = [\alpha_{\ell,i,j,p,q}, \alpha_{h,i,j,p,q}].$$

For the component constraint, we can perform a similar analysis. Suppose we are given two non-parallel data edges, each of which is designated by a base point $\mathbf{b}_i$ and an end point $\mathbf{e}_i$. These are chosen so that the tangent vector pointing from the base point to the end point is 90^o clockwise from the normal vector $\hat{\mathbf{n}}_i$ to the edge. For these two edges, we can compute the component of the vector $\mathbf{b}_j - \mathbf{b}_i$ in the direction of the normal vector $\hat{\mathbf{n}}_i$, which we call

$$d^{\perp}_{\ell,ij} = \langle \mathbf{b}_j - \mathbf{b}_i, \hat{\mathbf{n}}_i \rangle$$

and the component

$$d^{\perp}_{h,ij} = \langle \mathbf{e}_j - \mathbf{b}_i, \hat{\mathbf{n}}_i \rangle .$$

Then given a corresponding pair of model edges, we can compute similar components

$$M^{\perp}_{\ell,pq} = \left\langle \mathbf{B}_q - \mathbf{B}_p, \hat{\mathbf{N}}_p \right\rangle$$

and

$$M^{\perp}_{h,pq} = \left\langle \mathbf{E}_q - \mathbf{B}_p, \hat{\mathbf{N}}_p \right\rangle .$$

We also let

$$\sigma = \text{signum}\left\{ \left\langle (\mathbf{e}_i - \mathbf{b}_i)^{\perp}, (\mathbf{e}_j - \mathbf{b}_j) \right\rangle \right\}$$

and we let Δx_i and Δy_i denote the x and y components respectively of the vector $\mathbf{e}_i - \mathbf{b}_i$. Then the range of values of the stretch factor α is given by the range spanned by

$$\alpha = \frac{\sigma d^{\perp}_{\ell,ij} |\Delta y_i|}{\sqrt{(M^{\perp}_{\ell,pq})^2 - (d^{\perp}_{\ell,ij})^2 (\Delta x_i)^2}}$$

and

$$\alpha = \frac{\sigma d^{\perp}_{h,ij} \, |\Delta y_i|}{\sqrt{(M^{\perp}_{h,pq})^2 - (d^{\perp}_{h,ij})^2 \, (\Delta x_i)^2}}.$$

In fact, one must also allow for error in the measurements, which will yield a range of values for $d^{\perp}_{\ell,ij}$ and $d^{\perp}_{h,ij}$, leading to a larger range of values for the stretching factor α, again denoted

$$[\alpha_{\ell,i,j,p,q}, \alpha_{h,i,j,p,q}].$$

We define

$$\textbf{stretch-component-constraint}(i,j,p,q) = [\alpha_{\ell,i,j,p,q}, \alpha_{h,i,j,p,q}].$$

Finally, we can also alter the length constraint, which in this case is given by

$$\textbf{stretch-length-constraint}(i,p) = \left[\sqrt{\frac{(L_p - \epsilon_L)^2 - (\Delta x_i)^2}{(\Delta y_i)^2}}, \infty\right].$$

Given these unary and binary constraints, we can now modify our constrained search process. With each node of the search tree, we associate a range of consistent values for the stretch parameter, which we will denote $[\alpha^{(k)}_{\ell}, \alpha^{(k)}_{h}]$, where k indicates the level of the node in the tree. Suppose the search process is currently at some node at level k in the interpretation tree and with a consistent partial interpretation given by

$$\{(d_1, m_{j_1}), (d_2, m_{j_2}), \ldots (d_k, m_{j_k})\}.$$

We now consider the next data fragment d_{k+1}, and its possible assignment to model fragment $m_{j_{k+1}}$, where j_{k+1} varies from 1 to $n+1$.

The following rules hold.

- If $m_{j_{k+1}}$ is the wild card match, then the new interpretation

  $$\{(d_1, m_{j_1}), (d_2, m_{j_2}), \ldots (d_{k+1}, m_{j_{k+1}})\}$$

  is consistent, and we continue downward in our search, setting

  $$[\alpha^{(k+1)}_{\ell}, \alpha^{(k+1)}_{h}] = [\alpha^{(k)}_{\ell}, \alpha^{(k)}_{h}].$$

- If $m_{j_{k+1}}$ is a linear edge segment, we let

  $$[\alpha^{(k+1)}_{\ell}, \alpha^{(k+1)}_{h}] = [\alpha^{(k)}_{\ell}, \alpha^{(k)}_{h}] \bigcap \textbf{stretch-length-constraint}(k+1, j_{k+1}).$$

  If this new range is non-empty, then for all $i \in \{1, \ldots, k\}$ such that d_i is a linear edge fragment, we let

  $$\begin{aligned}[\alpha^{(k+1)}_{\ell}, \alpha^{(k+1)}_{h}] = &[\alpha^{(k+1)}_{\ell}, \alpha^{(k+1)}_{h}] \\ &\bigcap \textbf{stretch-component-constraint}(i, k+1, j_i, j_{k+1}) \\ &\bigcap \textbf{stretch-component-constraint}(k+1, i, j_{k+1}, j_i) \\ &\bigcap \textbf{binary-angle-constraint}(i, k+1, j_i, j_{k+1}).\end{aligned}$$

- If

  $$[\alpha^{(k+1)}_{\ell}, \alpha^{(k+1)}_{h}]$$

is non-empty, then

$$\{(d_1, m_{j_1}), (d_2, m_{j_2}), \ldots (d_{k+1}, m_{j_{k+1}})\}$$

is a consistent partial interpretation, and we continue our depth first search. Otherwise, the partial interpretation is inconsistent. In this case, we increment the model fragment index j_{k+1} by 1 and try again, until $j_{k+1} = n + 1$.

If the search process is currently at some node at level k in the interpretation tree, and has an inconsistent partial interpretation given by

$$\{(d_1, m_{j_1}), (d_2, m_{j_2}), \ldots (d_k, m_{j_k})\}$$

then it is in the process of backtracking. If $j_k = n + 1$ (the wild card) we backtrack up another level, otherwise we increment j_k and continue.

Figure 5 shows an example of a set of overlapping handles (taken from the family illustrated in Figure 1). Each instance of one of the handles is identified and located, including determining the actual value of the stretching parameter.

3.6 Combining Parameterizations

It is useful to be able to recognize objects that combine different types of parameterizations. For example, consider a pair of shears, that have both a rotational freedom between the two blades, and a stretching freedom along the axis of each blade. We can combine the methods described in Sections 3.3 and 3.4 to deal with this more general problem. An example is shown in Figures 6–9. Here, the system correctly solves for the position and orientation of the object, the angle of rotation between the blades and the stretching factor of the blades.

One could also combine stretching and scaling parameters in a single family of objects. This is equivalent to allowing independent stretching in two orthogonal directions. In this case, there are two parameters for which to solve, so that each constraint only specifies a relationship between the parameters. We can define a two-dimensional parameter space, spanned by the stretching parameter in the x and y directions. Initially, this space will contain a region of feasibility, defined by any limits on the range of parameters. As we add each constraint in our search process, a new region of the space will be defined, and the intersection of the two will determine the range of feasible parameter values consistent with the current interpretation. As in the earlier cases, if the region of feasibility becomes empty, the interpretation is inconsistent.

Determining the region of feasibility defined by the constraints is somewhat delicate. The length constraint, for example, yields an ellipse centered at the origin of the parameter space, whose complement demarks the feasible region. The angle constraint yields a feasible region that consists of a contiguous family of rays passing through the origin of the parameter space. In principle, one could use such constraints, together with procedures for intersecting regions in the plane to implement a recognition system for parts that stretch, scale and rotate at the same time. We have not yet done so.

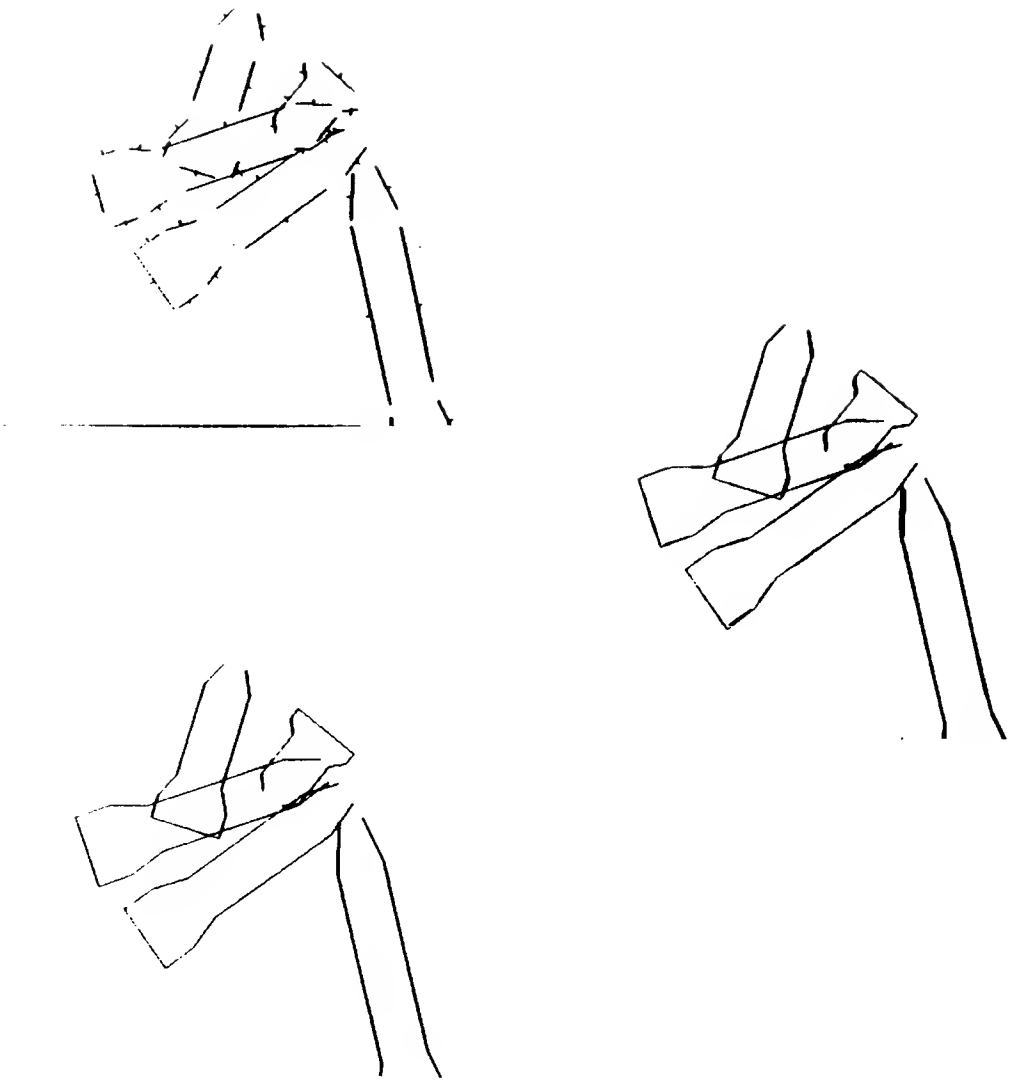

Figure 5. Examples of recognition when the free parameter is stretching along an axis. The first part shows a set of linear edges segments, the second shows the overlay of the located objects, and the third shows the located objects in isolation.

One drawback of the system presented here is that the analysis of how to parameterize an object model is done by hand, rather than automatically determining the parameterization for a model [Brooks 81].

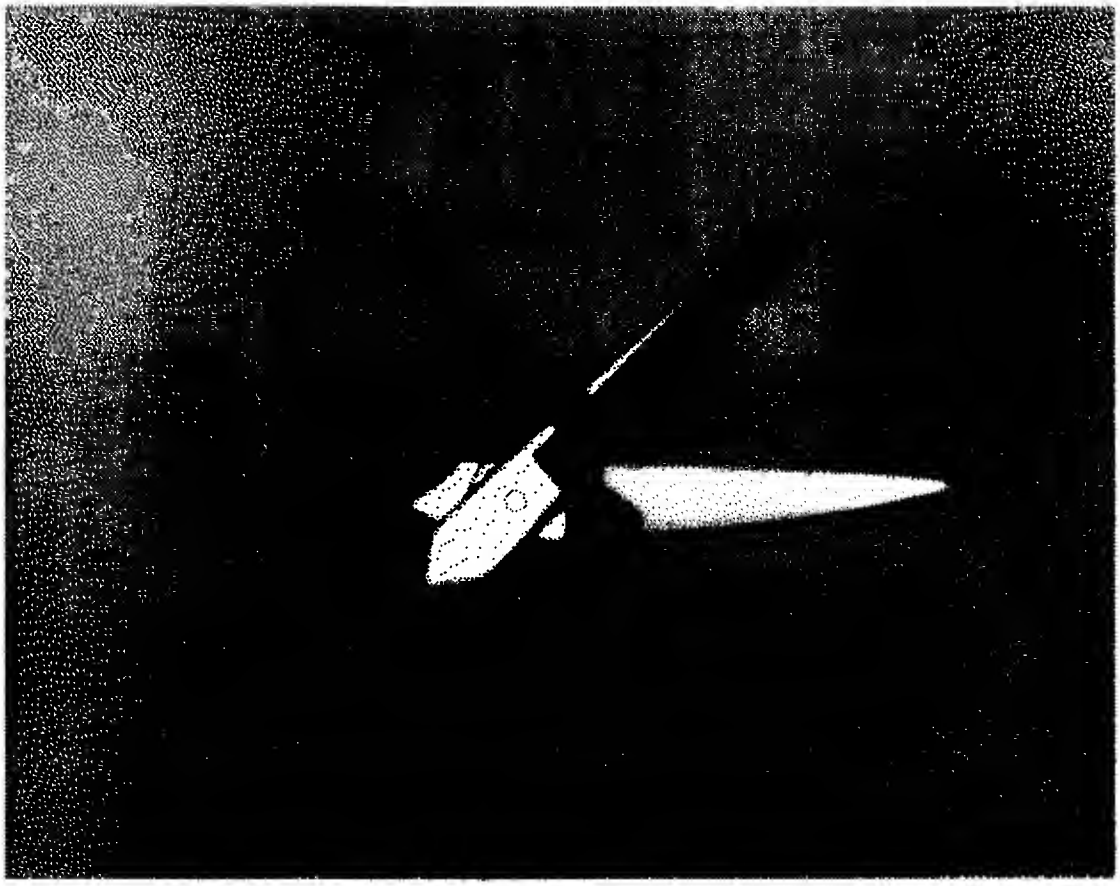

Figure 6. Example of recognition with different types of parameterizations. The object has a rotational free parameter and a stretching free parameter. The figure shows the original image.

4. Relation to previous work

The literature on object recognition stretches over a period of at least twenty years. An extensive (70 page) review of much of this literature for 3D objects can be found in [Besl and Jain 1985]. A survey of model-based image analysis systems can be found in [Binford 1982].

In terms of the approach to be described here, a number of authors have taken a similar view to ours that recognition can be structured as an explicit search for a match between data elements and model elements [Ayache and Faugeras 86, Baird 85, Bolles and Cain 82, Bolles, Horaud and Hannah 83, Faugeras and Hebert 83, Goad 83, Ikeuchi 87, Lowe 86, Stockman and Esteva 84]. Of these, the work of Bolles and his colleagues, Faugeras and his colleagues, and that of Baird are closest to the approach presented here.

The interpretation tree approach is an instance of the consistent labeling problem that has been studied extensively in computer vision and artificial intelligence [Waltz 75, Montanari 74, Mackworth 77, Freuder 78, 82, Haralick and Shapiro 79,

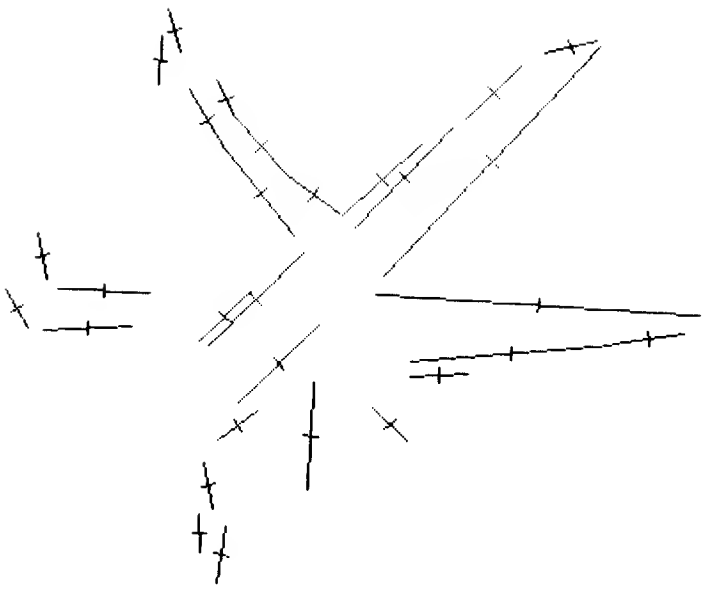

Figure 7. Example of recognition with different types of parameterizations. The edge fragments extracted from the image in Figure 6 are shown.

Haralick and Elliott 80, Mackworth and Freuder 85]. This paper can be viewed as suggesting a particular consistency relation (the constraints on distances and angles) and exploring its performance in a wide variety of circumstances. An alternative approach to the solution of consistent labeling problems is the use of relaxation. A number of authors have investigated this approach to object recognition [Davis 79, Bhanu and Faugeras 84, Ayache and Faugeras 82]. These techniques are more suitable for implementation on parallel machines.

The literature on recognition of parameterized objects is much smaller. The best known system is probably ACRONYM [Brooks 81], which also attacks the recognition problem by means of constraints to reduce ranges of parameterized variables. One of the main differences is that Brooks' system dealt with both rigid subparts and constraints that incorporated free parameters at runtime. In the approach presented here, we are compiling special cases of parameterization by hand in advance, so that the runtime portion of the problem is much simpler, and uses stronger constraints. This makes our system somewhat less general than Brooks', although it does benefit from a simpler recognition engine.

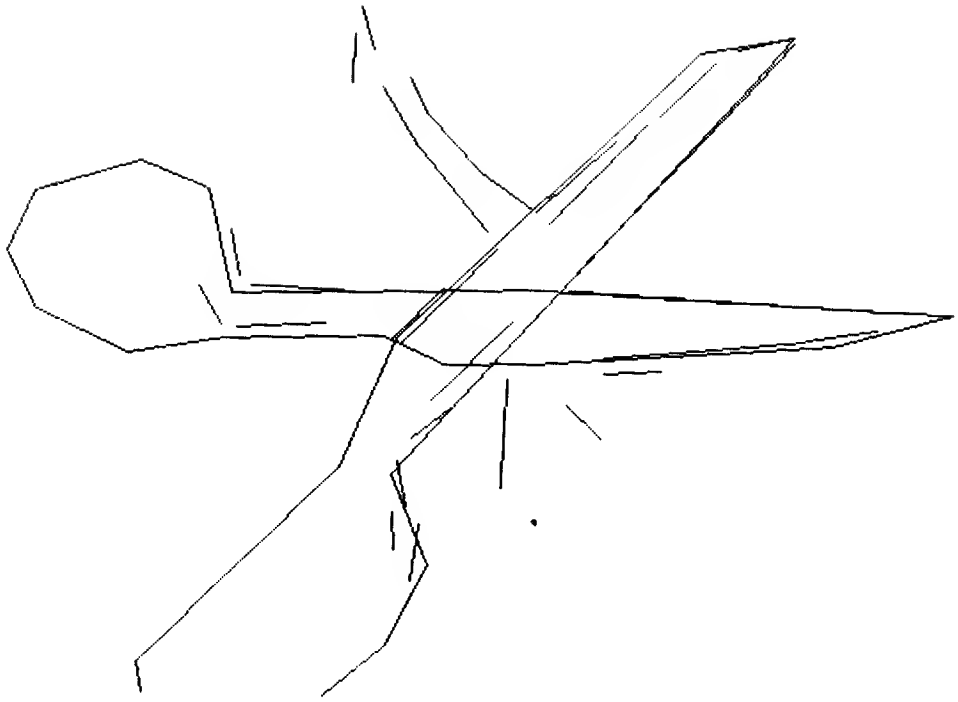

Figure 8. Example of recognition with different types of parameterizations. The solution is overlaid on the edge fragments extracted from the image in Figure 6.

Acknowledgements

The author would like to thank R. Bolles, R. Brooks, O. Faugeras, B. Horn, and T. Lozano-Pérez for useful comments and suggestions.

References

Ayache, N. J. and Faugeras, O. D. 1982. Recognition of partially visible planar shapes. *Proc. 6th Intl. Conf. Pattern Recognition*, Munich.

Ayache, N. J. and Faugeras, O. D. 1986. HYPER: A new approach for the recognition and positioning of two-dimensional objects. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-8(1):44–54.

Baird, H. 1986. *Model-Based Image Matching Using Location.* Cambridge: MIT Press.

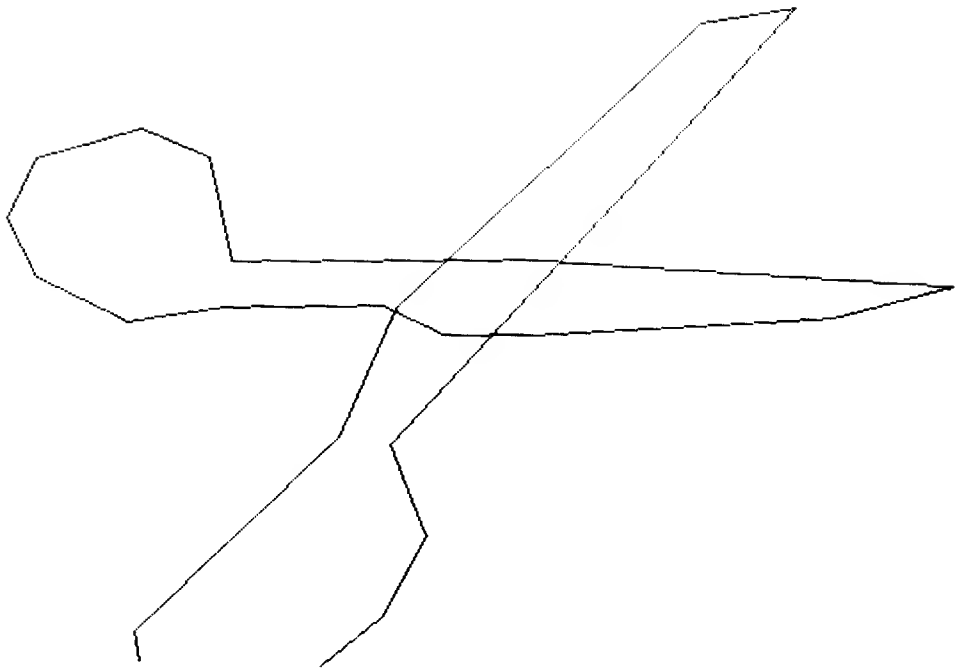

Figure 9. Example of recognition with different types of parameterizations. The solution for the image in Figure 6 is shown in isolation.

Ballard, D. H. and Brown, C. M. 1982. *Computer Vision.* Englewood Cliffs:Prentice Hall.

Besl, P. J. and Jain, R. C. 1985. Three-Dimensional Object Recognition. *ACM Computing Surveys* 17(1): 75–145.

Bhanu, B. and Faugeras, O. D. 1984. Shape matching of two-dimensional objects. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-6(3).

Binford, T. O. 1982. Survey of model-based image analysis systems. *Int. Journ. of Robotics Research* 1(1):18–64.

Bolles, R. C., and Cain, R. A. 1982. Recognizing and locating partially visible objects: The Local-Feature-Focus method. *Int. J. Robotics Res.* 1(3):57–82.

Bolles, R. C., Horaud, P., and Hannah, M. J. 1983. 3DPO: A three-dimensional part orientation system. Paper delivered at First International Symposium of Robotics Research, Bretton Woods, N.H. (Also in Robotics Research: The First International Symposium, edited by M. Brady and R. Paul, MIT Press, 1984, pp. 413–424.)

Brooks, R., 1981. Symbolic reasoning among 3-dimensional models and 2-dimensional images. *Artificial Intel.* 17:285–349.

Davis, L. Shape matching using relaxation techniques. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-1(1):60–72.

Drumheller, M. 1987. Mobile Robot Localization Using Sonar. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-9(2): 325–332. (See also: S. B. Thesis, Dept. of Mechanical Engineering, MIT, 1984 and MIT AI Lab Memo 826, Mobile Robot Localization Using Sonar.)

Faugeras, O. D. and Hebert, M. 1983 (Aug., Karlsruhe, W. Germany). A 3-D recognition and positioning algorithm using geometrical matching between primitive surfaces. *Proc. Eighth Int. Joint Conf. Artificial Intell.* Los Altos: William Kaufmann, pp. 996–1002.

Faugeras, O. D., Hebert, M., and Pauchon, E. 1983 (June, Washington DC). Segmentation of range data into planar and quadratic patches. *Proc. CVPR'83.*

Freuder, E. C. 1978. Synthesizing constraint expressions. *Comm. of the ACM*, 21(11), pp. 958–966.

Freuder, E. C. 1982. A sufficient condition for backtrack-free search. *J. ACM*, 29(1), pp. 24–32.

Gaston, P. C., and Lozano-Pérez, T. 1984. Tactile recognition and localization using object models: The case of polyhedra on a plane. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-6(3):257–265.

Goad, C. 1983. Special purpose automatic programming for 3d model-based vision. in *Proceedings of DARPA Image Understanding Workshop.*

Grimson, W. E. L., 1986. The combinatorics of local constraints in model-based recognition and localization from sparse data. *J. ACM* 33(4):658–686.

Grimson, W. E. L., 1987. On the recognition of curved objects. *MIT AI Lab Memo* 983, 1987.

Grimson, W. E. L., and Lozano-Pérez, T. 1984. Model-based recognition and localization from sparse range or tactile data. *Int. J. Robotics Res.* 3(3):3–35.

Grimson, W. E. L. and Lozano-Pérez, T. 1987. Localizing overlapping parts by searching the interpretation tree. *IEEE Trans. Patt. Anal. and Mach. Intel.* PAMI-9(4):469–482. (see also MIT AI Lab Memo 841, June 1985.)

Haralick, R. M. and Elliott, G. 1980. Increasing tree search efficiency for constraint satisfaction problems. *Artificial Intelligence*, Vol. 14, pp. 263–313.

Haralick, R. M. and Shapiro, L. G. 1979. The consistent labeling problem: Part I. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-1(4):173–184.

Ikeuchi, K. 1987. Generating an interpretation tree from a CAD model for 3d-object recognition in bin-picking tasks. *Int. Journ. Computer Vision*, Vol. 1, No. 2, pp. 145–166.

Lowe, D. G. 1986. Three-dimensional object recognition from single two-dimensional images. Courant Institute Robotics Report, No. 62.

Marr, D. and Hildreth, E. C. 1980. Theory of edge detection. *Proc. R. Soc. Lond. B* 207:187–217.

Mackworth, A. K. 1977. Consistency in networks of constraints. *Artificial Intelligence*, Vol. 8, pp. 99–118.

Mackworth, A. K. and Freuder, E. C. 1985. The complexity of some polynomial network consistency algorithms for constraint satisfaction problems. *Artificial Intelligence*, Vol. 25, pp. 65–74.

Montanari, U. 1974. Networks of constraints: Fundamental properties and applications to picture processing. *Inform. Sci.*, Vol. 7, pp 95–132.

Stockman, G., and Esteva, J. C. 1984. Use of geometrical constraints and clustering to determine 3D object pose. TR84-002. East Lansing, Mich.:Michigan State University Department of Computer Science.

Waltz, D. 1975. Understanding line drawings of scenes with shadows. in *The Psychology of Computer Vision*, P. Winston, Ed. New York:Mc Graw Hill, pp 19 – 91.

CS-TR Scanning Project
Document Control Form

Date : 5/26/95

Report # AIM-985

Each of the following should be identified by a checkmark:
Originating Department:

☒ Artificial Intellegence Laboratory (AI)
☐ Laboratory for Computer Science (LCS)

Document Type:

☐ Technical Report (TR) ☒ Technical Memo (TM)
☐ Other:_____

Document Information

Number of pages: 26 (30 - IMAGES)
Not to include DOD forms, printer instructions, etc... original pages only.

Originals are:
☒ Single-sided or
☐ Double-sided

Intended to be printed as :
☐ Single-sided or
☒ Double-sided

Print type:
☐ Typewriter ☐ Offset Press ☒ Laser Print
☐ InkJet Printer ☐ Unknown ☐ Other:_____

Check each if included with document:

☐ DOD Form ☐ Funding Agent Form ☐ Cover Page
☐ Spine ☐ Printers Notes ☐ Photo negatives
☐ Other: _____

Page Data:

Blank Pages (by page number): _____

Photographs/Tonal Material (by page number): 20

Other (note description/page number):

Description :	Page Number:
(A) IMAGE MAP (1) UN#'ED TITLE PAGE	
(26) PAGES #'ED 1-25	
(27) SCANCONTROL	
(28-30) TRGTS (3)	
(B) CUT & PASTE FIGS ON PAGES	13, 14, 19-23

Scanning Agent Signoff:
Date Received: 5/26/95 Date Scanned: 6/5/95 Date Returned: 6/8/95

Scanning Agent Signature: Michael W. Cook

Scanning Agent Identification Target

Scanning of this document was supported in part by the **Corporation for National Research Initiatives,** using funds from the **Advanced Research Projects Agency** of the **United states Government** under Grant: **MDA972-92-J1029.**

The scanning agent for this project was the **Document Services** department of the **M.I.T Libraries.** Technical support for this project was also provided by the **M.I.T. Laboratory for Computer Sciences.**